Least-squares reverse-time migration via deep learning-based updating operators

Kristian Torres

Supervisor: Dr. Mauricio Sacchi

Signal Analysis and Imaging Group (SAIG)
Department of Physics
University of Alberta

January 25, 2022

Outline

1. Least-squares migration (LSM)
2. Deep learning-based LSRTM
   - Learned iterative reconstruction
   - Learned post-processing operator
3. Numerical Experiments
4. Conclusions

Content

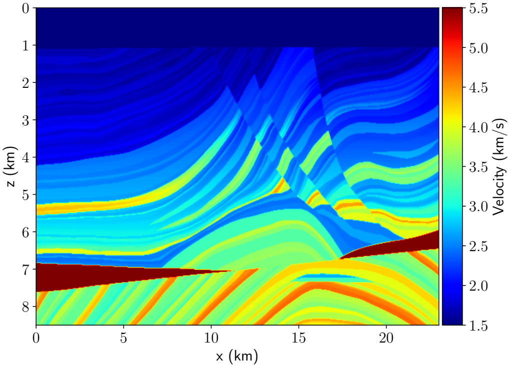
- **Acoustic wave equation**

$$\left(\frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right) u = f$$

c: velocity field

u: wavefield

f: source function



c

SIGNAL ANALYSIS & IMAGING GROUP

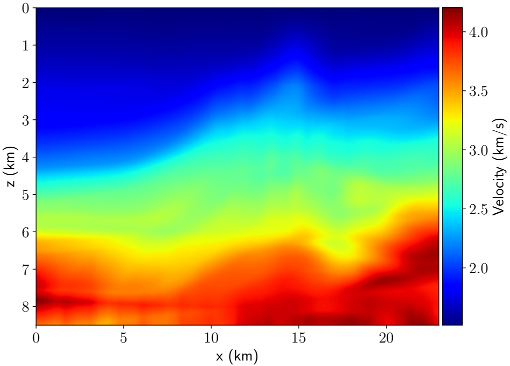
- **Acoustic wave equation**

$$\left(\frac{1}{c_0^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right) u_0 = f$$

c_0: velocity field

u_0: incident wavefield

f: source function



c_0

SIGNAL ANALYSIS & IMAGING GROUP

- **Acoustic wave equation**

$$\left(\frac{1}{c_0^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right)u_0 = f$$

c_0: velocity field

u_0: incident wavefield

f: source function

- **Acoustic wave equation**

$$\left(\frac{1}{c_0^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right) u_0 = f$$

c_0: velocity field

u_0: incident wavefield

f: source function

- **Linearized wave equation**

$$\left(\frac{1}{c_0^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right) \delta u = -m\frac{\partial^2 u_0}{\partial t^2}$$

c_0: velocity field

δu: scattered wavefield

m: reflectivity image

- **Acoustic wave equation**

$$\left(\frac{1}{c_0^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right) u_0 = f$$

c_0: velocity field

u_0: incident wavefield

f: source function

- **Linearized wave equation**

$$\left(\frac{1}{c_0^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right) \delta u = -m\frac{\partial^2 u_0}{\partial t^2}$$

c_0: velocity field

δu: scattered wavefield

m: reflectivity image

Assumption: No multiple scattering!

SIGNAL ANALYSIS & IMAGING GROUP

- **Forward modeling**

$$\mathbf{d} = \mathbf{Lm}$$

$\mathbf{d}$: seismic data

$\mathbf{L}$: forward modeling operator

$\mathbf{m}$: reflectivity image

Introduction

- **Forward modeling**

$$\mathbf{d} = \mathbf{L}\mathbf{m}$$

$\mathbf{d}$: seismic data

$\mathbf{L}$: forward modeling operator

$\mathbf{m}$: reflectivity image

- **Reverse time migration (RTM)**

$$\mathbf{m}_{\text{mig}} = \mathbf{L}^T \mathbf{d}_{\text{obs}}$$

$\mathbf{m}_{\text{mig}}$: RTM image

$\mathbf{L}^T$: migration operator

$\mathbf{d}_{\text{obs}}$: observed seismic data

- **Forward modeling**

$$\mathbf{d} = \mathbf{L}\mathbf{m}$$

$\mathbf{d}$: seismic data

$\mathbf{L}$: forward modeling operator

$\mathbf{m}$: reflectivity image

- **Reverse time migration (RTM)**

$$\mathbf{m}_{\text{mig}} = \mathbf{L}^T \mathbf{d}_{\text{obs}}$$

$\mathbf{m}_{\text{mig}}$: RTM image

$\mathbf{L}^T$: migration operator

$\mathbf{d}_{\text{obs}}$: observed seismic data

Migration is an approximate solution of the linearized inverse problem

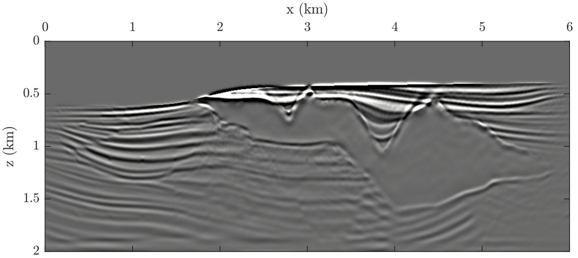
SIGNAL ANALYSIS & IMAGING GROUP

Least-squares migration (LSM)

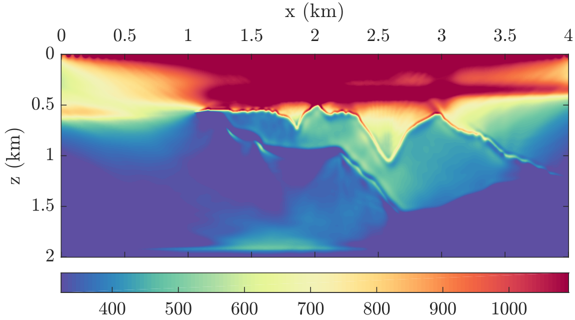
$$\mathbf{m}_{\mathrm{LSM}} = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{d}$$

Least-squares migration (LSM)

$$\mathbf{m}_{\text{LSM}} = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{d}$$



$$\mathbf{m}_{\text{RTM}} = \mathbf{L}^T\mathbf{d}$$

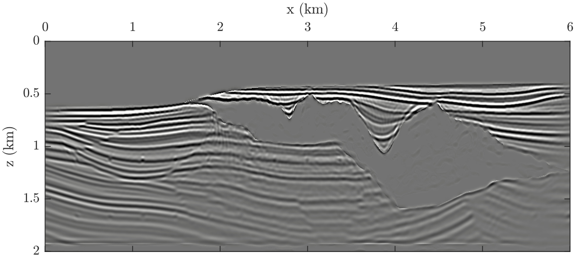


Illumination map

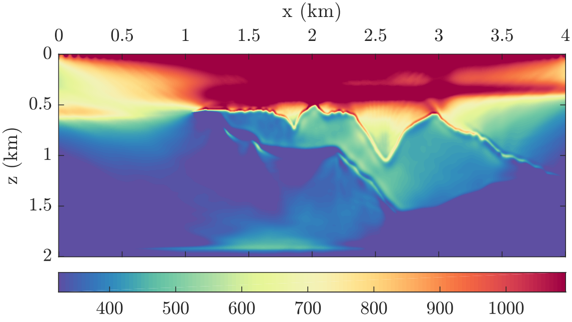
SIGNAL ANALYSIS & IMAGING GROUP

Least-squares migration (LSM)

$$\mathbf{m}_{\text{LSM}} = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{d}$$



$\mathbf{m}_{\text{LSM}}$

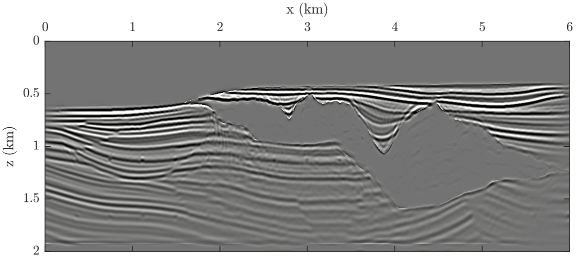


Illumination map

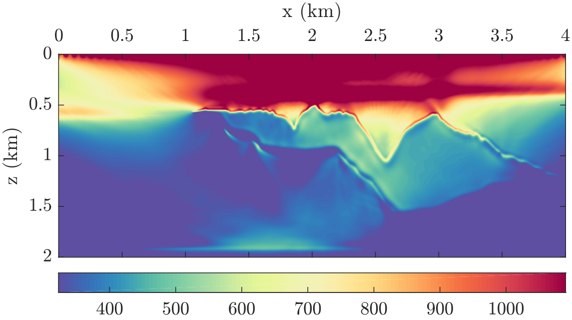
SIGNAL ANALYSIS & IMAGING GROUP

Least-squares migration (LSM)

$$\mathbf{m}_{\mathrm{LSM}} = \mathbf{H}^{-1}\mathbf{m}_{\mathrm{RTM}}$$



$\mathbf{m}_{\mathrm{LSM}}$



Illumination map

Solve unconstrained optimization problem iteratively

$$\min_{\mathbf{m}} \left\{ J(\mathbf{m}) + \lambda R(\mathbf{m}) = \frac{1}{2}||\mathbf{L}\mathbf{m} - \mathbf{d}_{\text{obs}}||_2^2 + \lambda R(\mathbf{m}) \right\},$$

Introduction

Solve unconstrained optimization problem iteratively

$$\min_{\mathbf{m}} \left\{ J(\mathbf{m}) + \lambda R(\mathbf{m}) = \frac{1}{2} ||\mathbf{L}\mathbf{m} - \mathbf{d}_{\text{obs}}||_2^2 + \lambda R(\mathbf{m}) \right\},$$

When $R(\mathbf{m})$ is differentiable: gradient-based method such as GD or CG

Solve unconstrained optimization problem iteratively

$$\min_{\mathbf{m}} \left\{ J(\mathbf{m}) + \lambda R(\mathbf{m}) = \frac{1}{2} ||\mathbf{L}\mathbf{m} - \mathbf{d}_{\text{obs}}||_2^2 + \lambda R(\mathbf{m}) \right\},$$

When $R(\mathbf{m})$ is differentiable: gradient-based method such as GD or CG

$$\text{GD solution} \rightarrow \mathbf{m}_{k+1} = \mathbf{m}_k - \alpha \big(\nabla J(\mathbf{m}_k) + \lambda \nabla R(\mathbf{m}_k) \big)$$

$$\nabla J(\mathbf{m}_k) = \mathbf{L}^T (\mathbf{L}\mathbf{m}_k - \mathbf{d}))$$

Alternatively, introduce prior knowledge as projected constraints:

$$\min_{\mathbf{m}\in\mathcal{C}} \left\{ J(\mathbf{m}) = \frac{1}{2}||\mathbf{Lm} - \mathbf{d}_{\text{obs}}||_2^2 \right\},$$

$\mathcal{C}$: set of desired physical constraints

Alternatively, introduce prior knowledge as projected constraints:

$$\min_{\mathbf{m}\in\mathcal{C}} \left\{ J(\mathbf{m}) = \frac{1}{2}||\mathbf{Lm} - \mathbf{d}_{\text{obs}}||_2^2 \right\},$$

$\mathcal{C}$: set of desired physical constraints

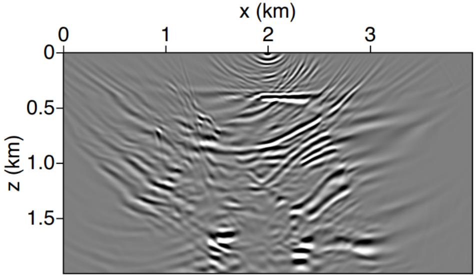
$$\text{Projected GD solution} \rightarrow \mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}\Big(\mathbf{m}_k - \alpha\nabla J(\mathbf{m}_k)\Big)$$

$$\text{PGD solution} \rightarrow \mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}\Big(\mathbf{m}_k - \alpha \nabla J(\mathbf{m}_k)\Big)$$

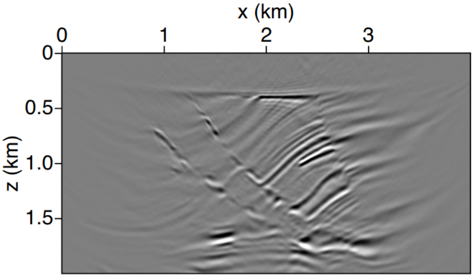
Least-squares migration via a gradient projection method - application to seismic data deblending (Cheng et al., 2016)

$$\text{PGD solution} \rightarrow \mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}\Big(\mathbf{m}_k - \alpha \nabla J(\mathbf{m}_k)\Big)$$

Least-squares migration via a gradient projection method - application to seismic data deblending (Cheng et al., 2016)



$\mathbf{m}_{\text{mig}}$



$\mathbf{m}_{\text{LSM}}$ via PGD

Classical LSM approaches

- Regularization/Constraints at each iteration: suppresses migration artifacts and improves inversion efficiency.

SIGNAL ANALYSIS & IMAGING GROUP

Classical LSM approaches

- Regularization/Constraints at each iteration: suppresses migration artifacts and improves inversion efficiency.
- Many iterations to find a good solution

SIGNAL ANALYSIS & IMAGING GROUP

Classical LSM approaches

- Regularization/Constraints at each iteration: suppresses migration artifacts and improves inversion efficiency.
- Many iterations to find a good solution
- Geology is too complicated $\rightarrow R(\mathbf{m})$? $\mathcal{P}_{\mathbb{C}}(\mathbf{m})$?

SIGNAL ANALYSIS & IMAGING GROUP

Classical LSM approaches

- Regularization/Constraints at each iteration: suppresses migration artifacts and improves inversion efficiency.
- Many iterations to find a good solution
- Geology is too complicated $\rightarrow R(\mathbf{m})$? $\mathcal{P}_{\mathbb{C}}(\mathbf{m})$?
- optimal λ?

Classical LSM approaches

- Regularization/Constraints at each iteration: suppresses migration artifacts and improves inversion efficiency.
- Many iterations to find a good solution
- Geology is too complicated $\rightarrow R(\mathbf{m})$? $\mathcal{P}_{\mathbb{C}}(\mathbf{m})$?
- optimal λ?

Deep Learning solutions for seismic imaging

Deep Learning solutions for inverse imaging

Main approaches

- Fully learned reconstruction (end-to-end)
- Learned iterative reconstruction
- Learned post-processing operator
- Learned regularizer
- Physics-informed (PINNs)
- Physics-guided based on RNNs
- Regularization via null space networks

Deep Learning solutions for inverse imaging

Main approaches

- Fully learned reconstruction (end-to-end)
- **Learned iterative reconstruction**
- Learned post-processing operator
- Learned regularizer
- Physics-informed (PINNs)
- Physics-guided based on RNNs
- Regularization via null space networks

Content

1. Least-squares migration (LSM)
2. Deep learning-based LSRTM
   - Learned iterative reconstruction
   - Learned post-processing operator
3. Numerical Experiments
4. Conclusions

Learned iterative reconstruction

Inspired by LSM via PGD

$$\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}\Big(\mathbf{m}_k - \alpha \nabla J(\mathbf{m}_k)\Big)$$

Inspired by LSM via PGD

$$\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}\Big(\mathbf{m}_k - \alpha \nabla J(\mathbf{m}_k)\Big)$$

Can we train updating operators to perform efficient projections?

Learned iterative reconstruction

Inspired by LSM via PGD

$$\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}\Big(\mathbf{m}_k - \alpha \nabla J(\mathbf{m}_k)\Big)$$

Can we train updating operators to perform efficient projections?

$$\mathbf{m}_{k+1} = \mathcal{P}_{\theta_k}\Big(\mathbf{m}_k - \alpha \nabla J(\mathbf{m}_k)\Big)$$

SIGNAL ANALYSIS & IMAGING GROUP

Inspired by LSM via PGD

$$\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}\Big(\mathbf{m}_k - \alpha \nabla J(\mathbf{m}_k)\Big)$$

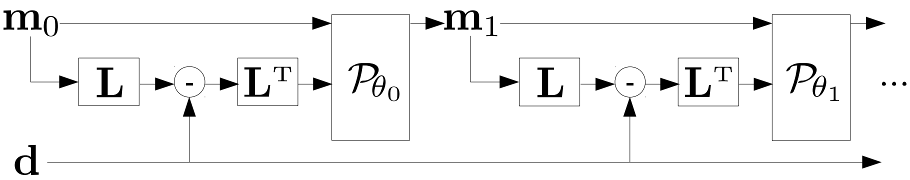
Can we train updating operators to perform efficient projections?

$$\mathbf{m}_{k+1} = \mathcal{P}_{\theta_k}\Big(\mathbf{m}_k - \alpha \nabla J(\mathbf{m}_k)\Big)$$

$$\boxed{\mathbf{m}_{k+1} = \mathcal{P}_{\theta_k}\Big(\mathbf{m}_k, \nabla J(\mathbf{m}_k)\Big)}$$

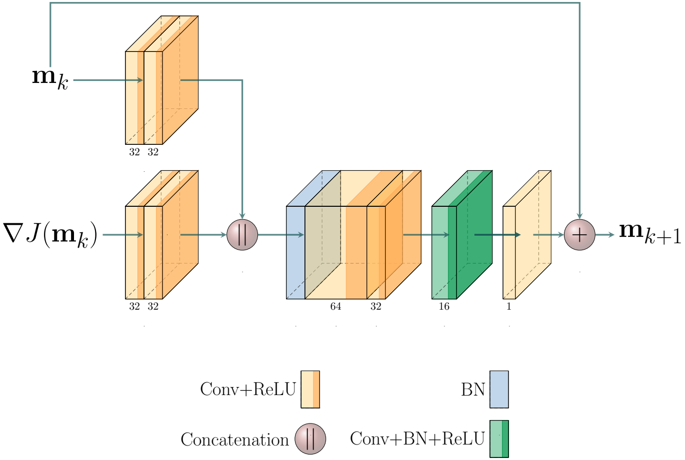
Deep-LSRTM

Unrolled algorithm (first two iterations)



Deep-LSRTM: network design

Each $\mathcal{P}_{\theta_k}$ is assembled following an encoder-decoder architecture:



Deep-LSRTM: training process

Instead of training all weights $\Theta = (\theta_0, ..., \theta_{K-1})$ together

$$\hat{\Theta} = \arg\min_{\Theta} \frac{1}{\Im} \sum_{i=1}^{\Im} ||\mathbf{m}_K^i - \mathbf{m}_{\text{true}}^i||_2^2,$$

$$= \arg\min_{\theta_0,...,\theta_{K-1}} \frac{1}{\Im} \sum_{i=1}^{\Im} ||(\mathcal{P}_{\theta_{K-1}} \circ ... \circ \mathcal{P}_{\theta_0}(\mathbf{m}_0^i, \nabla J(\mathbf{m}_0^i))) - \mathbf{m}_{\text{true}}^i||_2^2,$$

Deep-LSRTM: training process

Instead of training all weights $\Theta = (\theta_0, ..., \theta_{K-1})$ together

$$\hat{\Theta} = \arg\min_{\Theta} \frac{1}{\Im} \sum_{i=1}^{\Im} ||\mathbf{m}_K^i - \mathbf{m}_{\text{true}}^i||_2^2,$$

$$= \arg\min_{\theta_0,...,\theta_{K-1}} \frac{1}{\Im} \sum_{i=1}^{\Im} ||(\mathcal{P}_{\theta_{K-1}} \circ ... \circ \mathcal{P}_{\theta_0}(\mathbf{m}_0^i, \nabla J(\mathbf{m}_0^i))) - \mathbf{m}_{\text{true}}^i||_2^2,$$

Train by **greedy approach**:

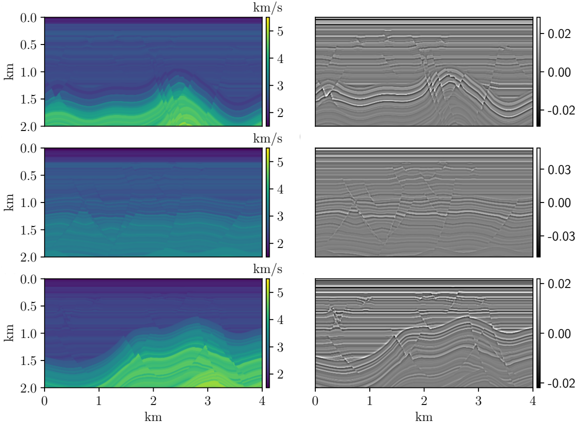
$$\hat{\theta}_k = \min_{\theta_k} \frac{1}{\Im} \sum_{i=1}^{\Im} ||\mathbf{m}_{k+1}^i - \mathbf{m}_{\text{true}}^i||_2^2$$

$$\min_{\theta_k} \frac{1}{\Im} \sum_{i=1}^{\Im} ||\mathcal{P}_{\theta_k}(\mathbf{m}_k^i, \nabla J(\mathbf{m}_k^i)) - \mathbf{m}_{\text{true}}^i||_2^2$$

$\Im \equiv$ No. of training instances

Data set

- Pseudo-random synthetic models
- 400×200 velocity distributions of sedimentary structures (1.5 to 5.5 km/s)
- Reflectivity as velocity perturbations: 900 training, 100 validation, 200 testing



Deep-LSRTM training

SIGNAL ANALYSIS & IMAGING GROUP

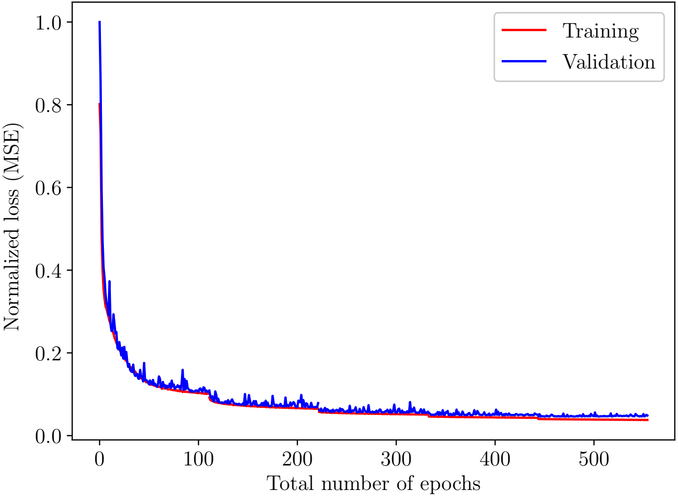
Each $\mathcal{P}_{\theta_k}$ is trained sequentially using

$\boxed{341,441 \text{ parameters}}$

- 50000 steps of Adam optimizer with lr $= 0.001$
- Batch size of 2, 111 epochs
- Including gradient calculation step, each updating operator is trained in ≈ 3 hours

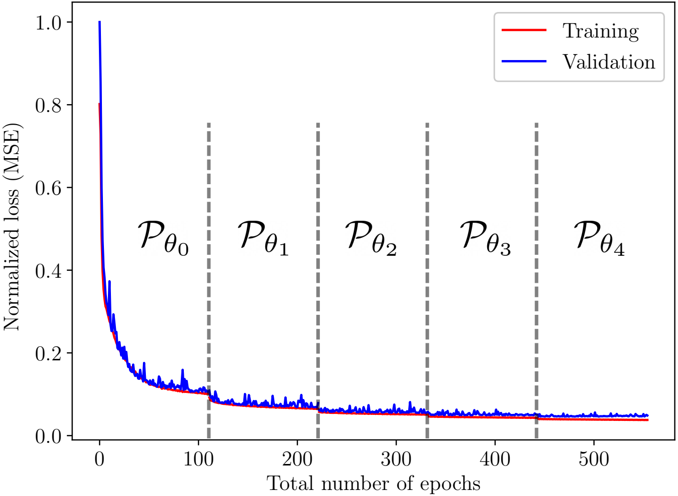
We set 5 iterations $(\mathcal{P}_{\theta_0}, ..., \mathcal{P}_{\theta_4})$ of Deep-LSRTM

Deep-LSRTM training



Deep-LSRTM training

SIGNAL ANALYSIS & IMAGING GROUP



Content

1. Least-squares migration (LSM)
2. Deep learning-based LSRTM
   - Learned iterative reconstruction
   - Learned post-processing operator
3. Numerical Experiments
4. Conclusions

SIGNAL ANALYSIS & IMAGING GROUP

Two-step reconstruction

$$\begin{aligned}\mathbf{m} &= \Lambda_{\Phi}(\mathbf{L}^{\mathrm{T}}\mathbf{d})\\ &= \Lambda_{\Phi}(\mathbf{m}_{\mathrm{RTM}})\end{aligned}$$

SIGNAL ANALYSIS & IMAGING GROUP

Two-step reconstruction

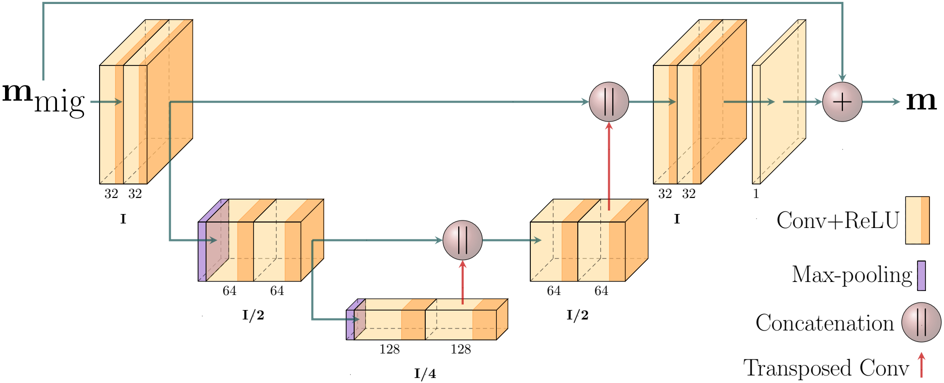
$$\begin{aligned} \mathbf{m} &= \Lambda_{\Phi}(\mathbf{L}^{\mathrm{T}}\mathbf{d}) \\ &= \Lambda_{\Phi}(\mathbf{m}_{\mathrm{RTM}}) \end{aligned}$$

Single-iteration image-domain LSM

$$\begin{aligned} \mathbf{m} &= \mathbf{C}\mathbf{L}^{\mathrm{T}}\mathbf{d} \\ &= \mathbf{C}\mathbf{m}_{\mathrm{mig}} \end{aligned}$$

$$\text{with } \mathbf{C} \approx \left[\mathbf{L}^{\mathrm{T}}\mathbf{L}\right]^{-1}$$

SIGNAL ANALYSIS & IMAGING GROUP

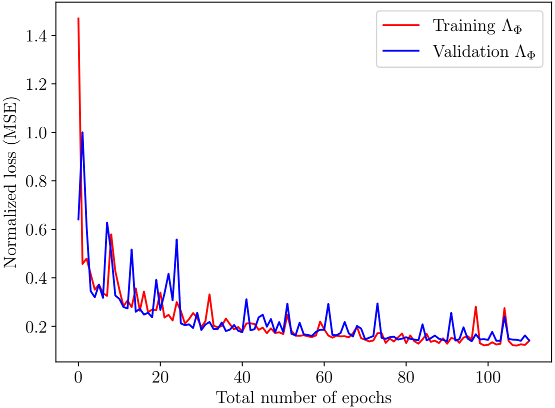


Modified U-net architecture

Single-step reconstruction with U-net

- Also trained with Adam (lr=0.001) for 111 epochs
- Modified version of the original U-net
- $\approx$ 3 hours to train

$\boxed{517,409 \text{ parameters}}$



Content

Numerical experiments

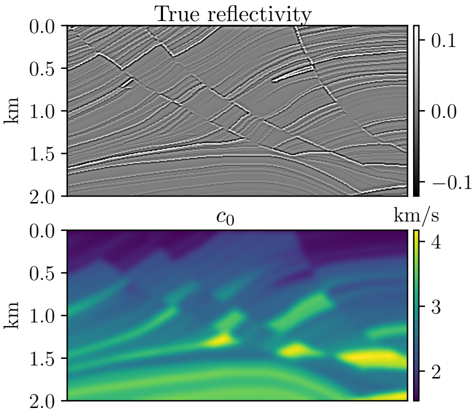
- Example 1: central part of Marmousi

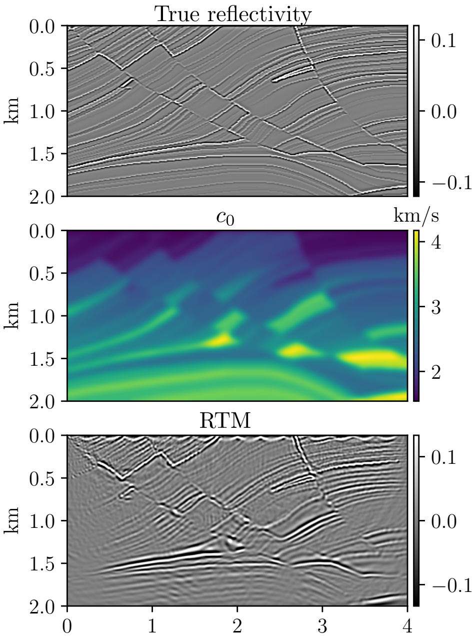
Numerical experiments

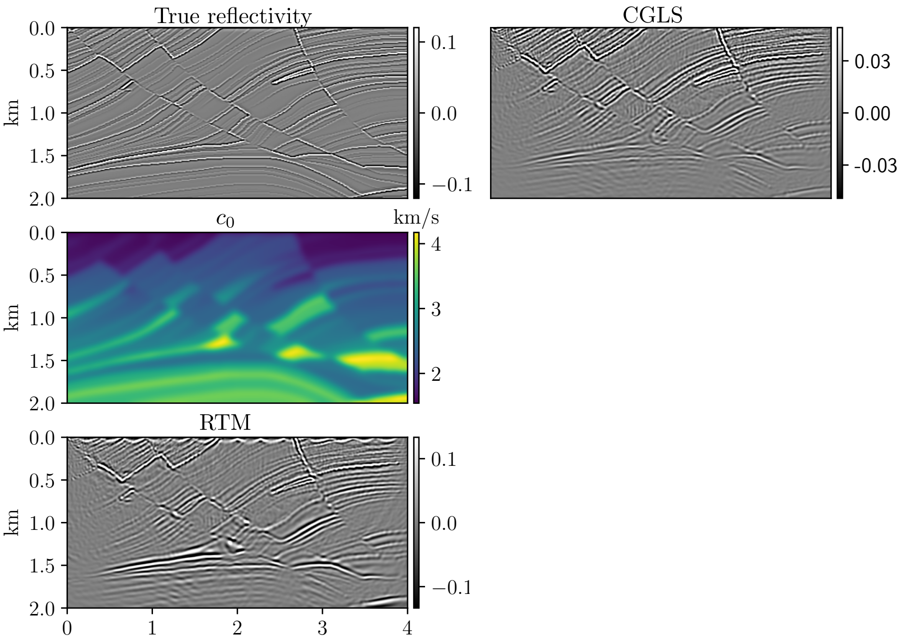
- Example 1: central part of 1Marmousi

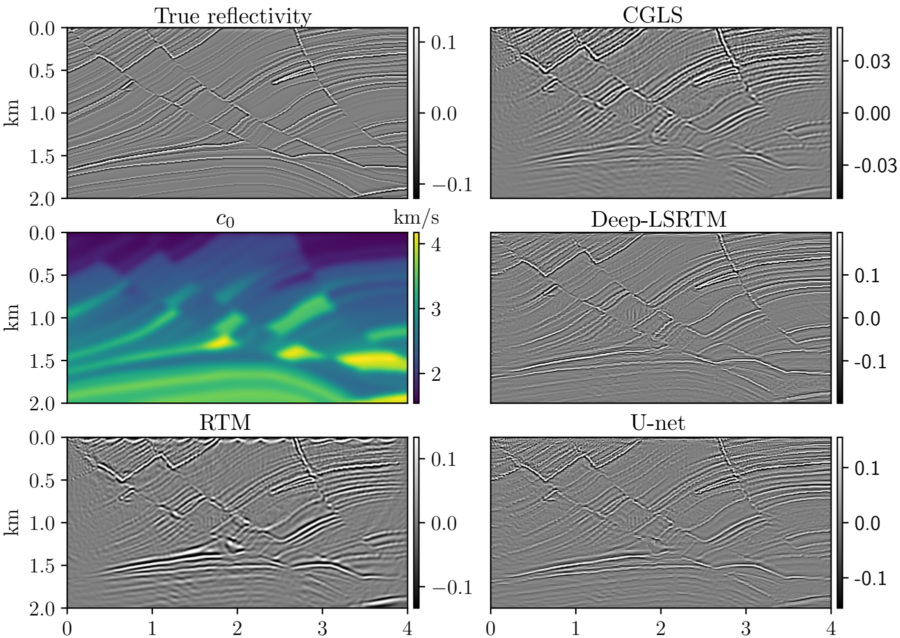
Numerical experiments

- Example 1: central part of Marmousi
  - Learned reconstructions as warm-starts for CGLS
  - Sensibility to background model errors
  - Sensibility to random noise

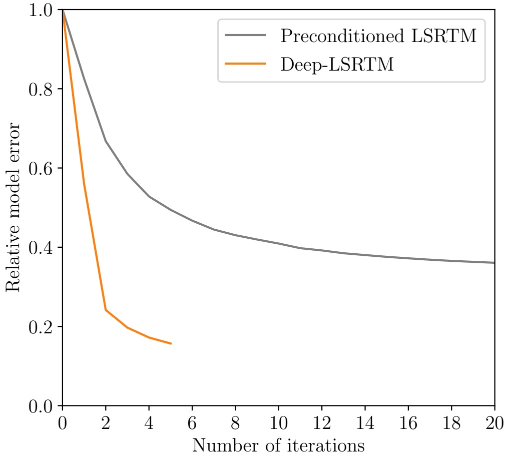

True reflectivity
0.0
0.5
1.0
1.5
2.0
km
0.1
0.0
−0.1
c_0
km/s
0.0
0.5
1.0
1.5
2.0
km
4
3
2


True reflectivity
km
0.0
0.5
1.0
1.5
2.0
0.1
0.0
−0.1
c_0
km/s
4
3
2
RTM
0
1
2
3
4


True reflectivity
CGLS
c_0
km/s
RTM
km
0.0
0.5
1.0
1.5
2.0
0.1
0.0
−0.1
0.03
0.00
-0.03
4
3
2
0
1
2
3
4


True reflectivity
CGLS
c_0
km/s
Deep-LSRTM
RTM
U-net
km
0.0
0.5
1.0
1.5
2.0
0
1
2
3
4
0.1
0.0
−0.1
0.03
0.00
-0.03
-0.1
2
3
4

Example 1: Marmousi model



$$\text{Relative model error } (\mathbf{m}_k) = \frac{||\mathbf{m}_k - \mathbf{m}_{\text{true}}||_2}{||\mathbf{m}_{\text{true}}||_2}$$

Numerical experiments

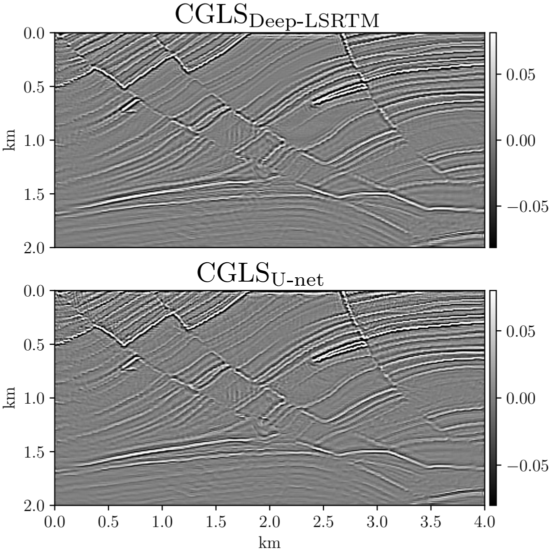
- Example 1: central part of Marmousi
    - Learned reconstructions as warm-starts for CGLS
    - Sensibility to background model errors
    - Sensibility to random noise

Example 1: Marmousi model

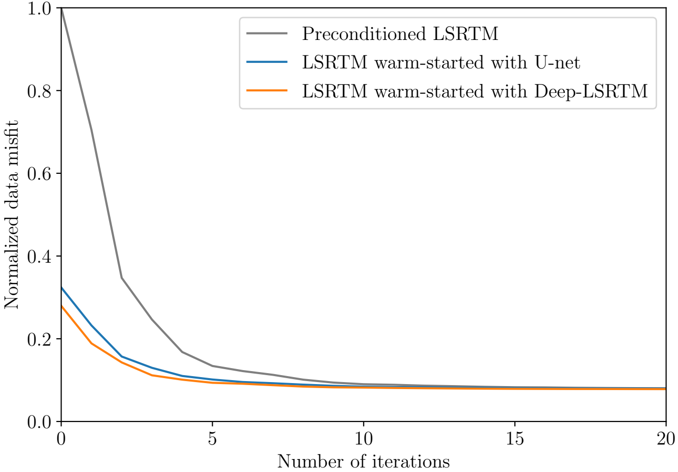
Cropped Marmousi		
Method	PSNR (db)	SSIM
CGLS	27.46	0.47
U-net	28.37	0.53
Deep-LSRTM	29.87	0.65
Warm-started CGLS		
$\text{CGLS}_{\text{U-net}}$	29.96	0.58
$\text{CGLS}_{\text{Deep-LSRTM}}$	30.37	0.69

Example 1: Marmousi model

Cropped Marmousi		
Method	PSNR (db)	SSIM
CGLS	27.46	0.47
U-net	28.37	0.53
Deep-LSRTM	29.87	0.65
Warm-started CGLS		
$\mathrm{CGLS}_{\text{U-net}}$	29.96	0.58
$\mathrm{CGLS}_{\text{Deep-LSRTM}}$	30.37	0.69



Example 1: Marmousi model



Numerical experiments

- Example 1: central part of Marmousi
  - Learned reconstructions as warm-starts for CGLS
  - Sensibility to background model errors
  - Sensibility to random noise

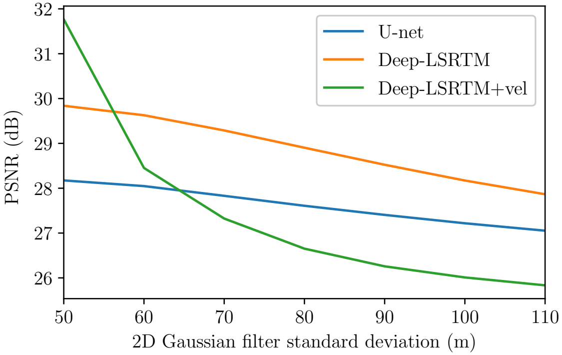
Example 1: Marmousi model

- Test learning approaches against background models with higher degrees of smoothing

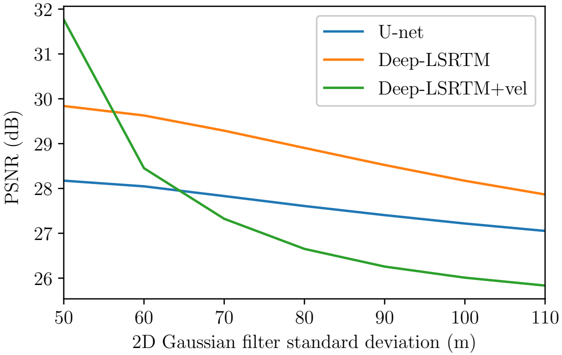
Example 1: Marmousi model

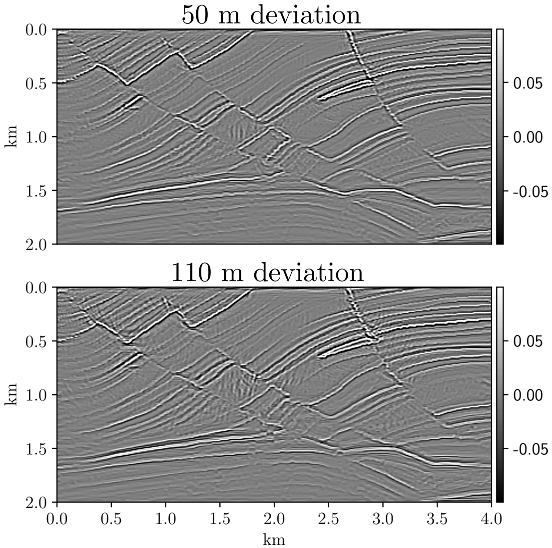
- Test learning approaches against background models with higher degrees of smoothing
- Deep-LSRTM+vel: background velocity field as complementary branch

- Test learning approaches against background models with higher degrees of smoothing
- Deep-LSRTM+vel: background velocity field as complementary branch

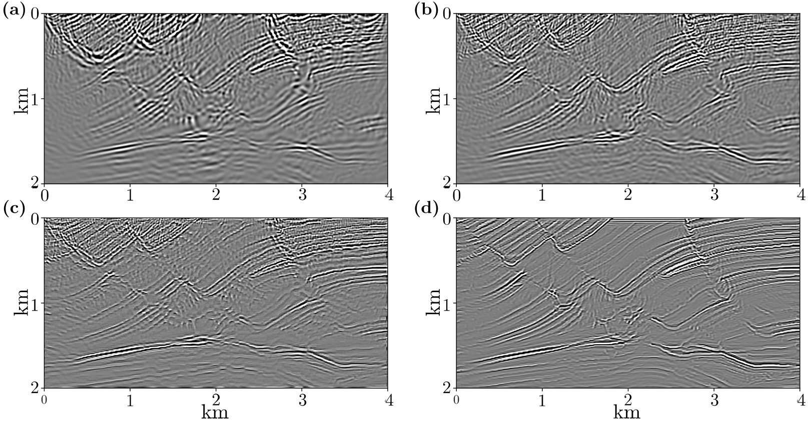


- Deep-LSRTM+vel





- Migration velocity model with 5% faster velocity everywhere.

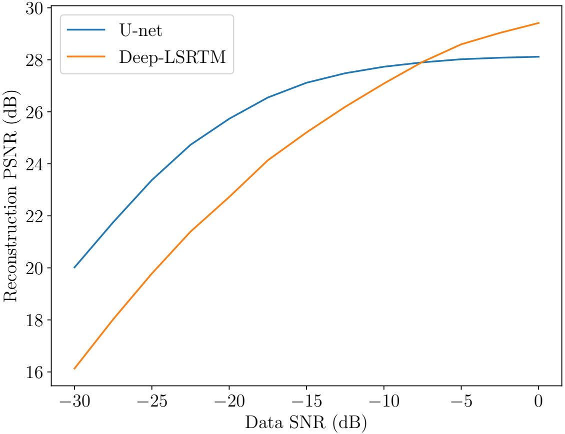


(a) RTM, (b) LSRTM (20 iterations), (c) U-net reconstruction, (d) Deep-LSRTM.

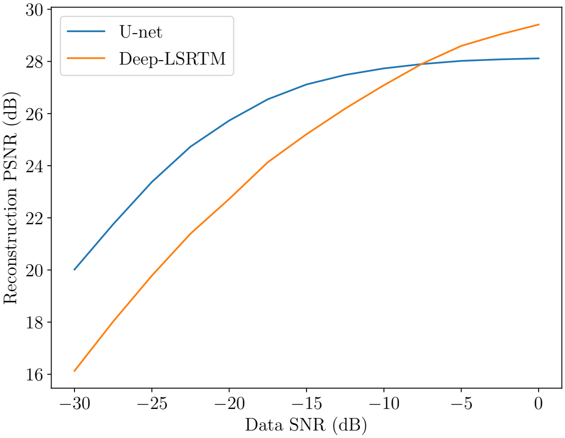
Numerical experiments

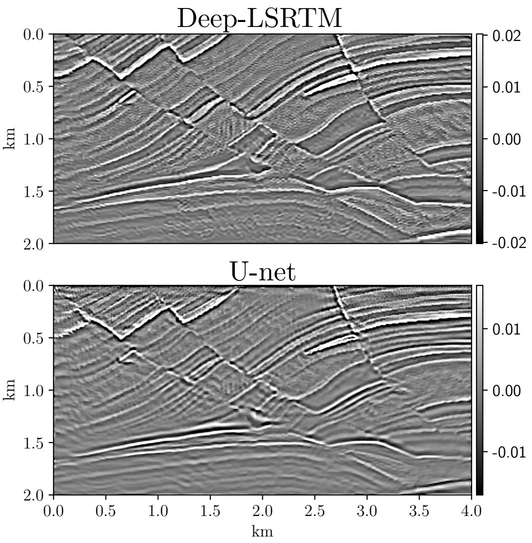
- Example 1: central part of Marmousi
  - Learned reconstructions as warm-starts for CGLS
  - Sensibility to background model errors
  - Sensibility to random noise

Example 1: Marmousi model



Example 1: Marmousi model





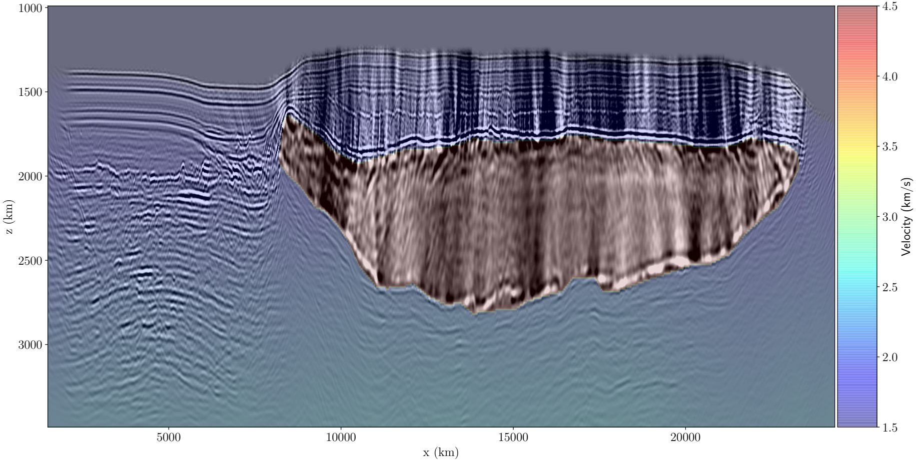
Example 2: Gulf of Mexico data set

SIGNAL ANALYSIS & IMAGING GROUP

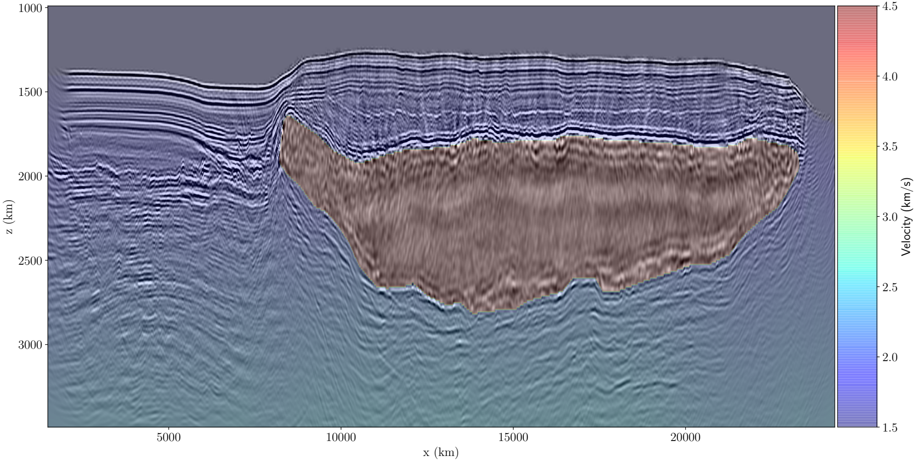
Mississippi Canyon (Gulf of Mexico) 2D data set

- Shallow salt body in a deep water environment
- The data lack both low frequencies and long offsets (maximum offset is 4.8 Km)
- Streamer geometry: 809 shots, 183 receivers, recording time = 7 s, dt = 4 ms.

Mississippi Canyon (Gulf of Mexico) data set



Mississippi Canyon (Gulf of Mexico) data set



Mississippi Canyon (Gulf of Mexico) data set

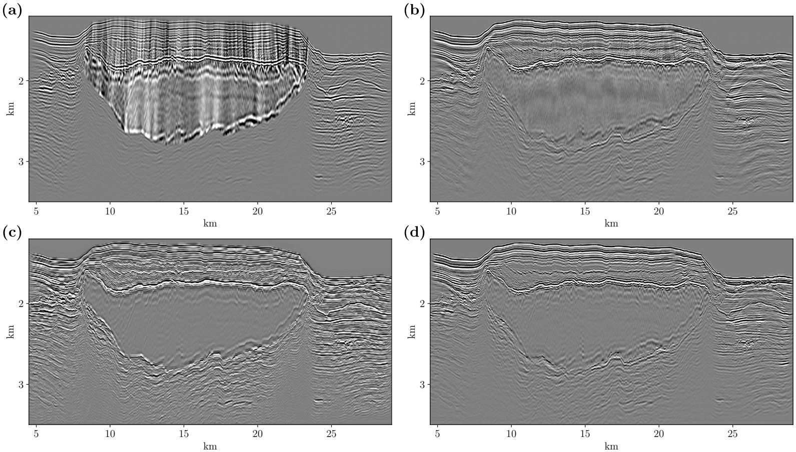
SIGNAL ANALYSIS & IMAGING GROUP

- **Challenges inherent to LSM**: accuracy of velocity model, salt body region, illumination issues, limited acquisition aperture, events not contained in the range of the forward (Born) operator, phase and amplitude corrections due to 3-D propagation.

- **Challenges inherent to application of supervised approach**: different wavelet (frequency content), different acquisition setup, different domain size, different distribution.

Example 2: Gulf of Mexico data set

SIGNAL ANALYSIS & IMAGING GROUP



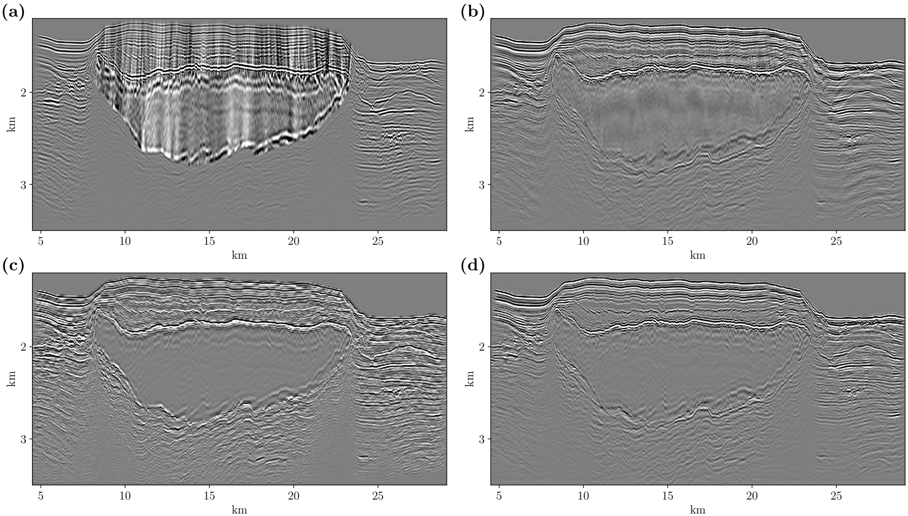
(a) RTM, (b) LSRTM (20 iterations), (c) Deep-LSRTM without transfer learning. (d) Deep-LSRTM **after transfer learning**.

Example 2: Gulf of Mexico data set

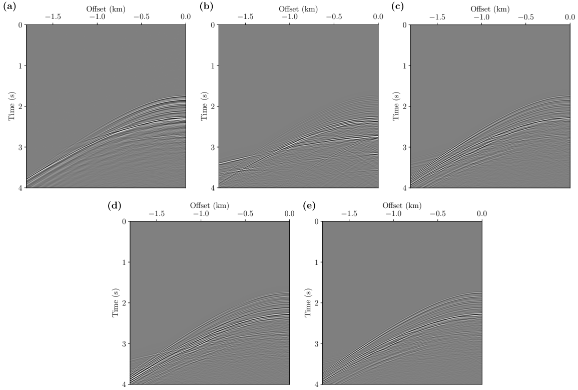
Transfer learning (re-training):

- Reference model: 20 iterations of preconditioned CGLS using a different group of only 60 shots.
- Retrain weights of each $\mathcal{P}_{\theta_k}$ with only 20 additional epochs and a reduced learning rate of 1e-5.

Example 2: Gulf of Mexico data set



QC: shot gather and demigration for source at x=15.6 km



(a) Observed gather, (b) RTM-demigrated, (c) LSRTM-demigrated, (d) Deep-LSRTM-demigrated (No transfer learning), (e) Deep-LSRTM-demigrated **after transfer learning**.

Content

Take-away

- Two CNN strategies to speed-up/improve seismic migration: iterative vs post-processing

Deep-LSRTM

$$\mathbf{m}_{k+1} = \Lambda_{\theta_k}(\mathbf{m}_k, \nabla J(\mathbf{m}_k))$$

SIGNAL ANALYSIS & IMAGING GROUP

- Two CNN strategies to speed-up/improve seismic migration: iterative vs post-processing

Deep-LSRTM

$$\mathbf{m}_{k+1} = \Lambda_{\theta_k}(\mathbf{m}_k, \nabla J(\mathbf{m}_k))$$

Two-step U-net reconstruction

$$\begin{aligned}\mathbf{m} &= \Lambda_{\Phi}(\mathbf{L}^{\mathrm{T}}\mathbf{d}) \\ &= \Lambda_{\Phi}(\mathbf{m}_{\mathrm{RTM}})\end{aligned}$$

Take-away

- Two CNN strategies to speed-up/improve seismic migration: iterative vs post-processing

Deep-LSRTM

$$\mathbf{m}_{k+1} = \Lambda_{\theta_k}(\mathbf{m}_k, \nabla J(\mathbf{m}_k))$$

Projected gradient descent LSM

$$\mathbf{m}_{k+1} = \mathcal{P}_{\mathbb{C}}\Big(\mathbf{m}_k - \alpha \nabla J(\mathbf{m}_k)\Big)$$

Two-step U-net reconstruction

$$\begin{aligned}\mathbf{m} &= \Lambda_{\Phi}(\mathbf{L}^{\mathrm{T}}\mathbf{d}) \\ &= \Lambda_{\Phi}(\mathbf{m}_{\mathrm{RTM}})\end{aligned}$$

Single-iteration image-domain LSM

$$\begin{aligned}\mathbf{m} &= \mathbf{C}\mathbf{L}^{\mathrm{T}}\mathbf{d} \\ &= \mathbf{C}\mathbf{m}_{\mathrm{RTM}}\end{aligned}$$

with $\mathbf{C} \approx \left[\mathbf{L}^{\mathrm{T}}\mathbf{L}\right]^{-1}$

Conclusions

- Despite using a small training set, the iterative Deep-LSRTM approach yields superior results than conventional LSRTM baselines for same No. of iterations.

- Deep-LSRTM also outperforms a two-step residual U-net post-migration application highlighting the value of including the forward and adjoint wave operators in the inference process.

- Deep-LSRTM network is not severely influenced by model over-fitting for synthetic tests. Re-training needed for Gulf of Mexico.

Thank you!

Least-squares reverse-time migration via deep learning-based updating operators

Kristian Torres

Supervisor: Dr. Mauricio Sacchi

Signal Analysis and Imaging Group (SAIG)
Department of Physics
University of Alberta

January 25, 2022

Deep-LSRTM: parametrization of the CNN blocks

SIGNAL ANALYSIS & IMAGING GROUP

$$\mathbf{m}_{k+1} = \Lambda_{\theta_k}\Big(\mathbf{m}_k, \nabla J(\mathbf{m}_k)\Big)$$

where

$$\Lambda_\theta = (\phi_N \circ W_{w_N,b_N}) \circ \ldots \circ (\phi_1 \circ W_{w_1,b_1}),$$

$$W^q_{w_n,b_n} = \Big(b^q_n + \sum_{p \in P} w^{q,p}_n * g_p\Big),\ \ q \in Q,$$

$$\theta = \Big((w_N, b_N), \ldots, (w_1, b_1)\Big),$$

and

$$\phi \leftarrow \text{ReLU, Sigmoid, Tanh, } \ldots$$

- Regularization effect and other parameters are learned implicitly
- No need to worry about learning data-to-model space mapping
- The data information is delivered through the gradient $\nabla J(\mathbf{m}_k, \mathbf{d})$

Figures of merit

- Quantitative comparison

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}}\right)$$

MAX_I: dynamic range
MSE: mean squared error

$$\mathrm{SSIM}(x, y) = \left[a_\mu(x, y)^\alpha \cdot c_\sigma(x, y)^\beta \cdot s_\sigma(x, y)^\gamma\right]$$

Amplitude: $a_\mu(x, y)$
Contrast: $c_\sigma(x, y)$
Structure: $s_\sigma(x, y)$